

**IMPACT OF AJAX DEVELOPMENT FRAMEWORK  
ON WEB 2**

**BY  
ALEX LEE HONG SIN**

Project Paper Submitted in Partial Fulfillment of the Requirement for  
the Degree of Master of Information Technology

Open University Malaysia  
(2007)

Digital Library OUM



0027829

CHAPTER	TABLE OF CONTENTS	PAGE
	ACKNOWLEDGEMENT	5
	ABSTRACT	6
	DECLARATION	7
1	INTRODUCTION	8
	1.1 Background of the Study	8
	1.2 Problem Statement	8
	1.3 Research Objectives	9
2	LITERATURE REVIEW ON WEB 2.0	9
	2.1 Introduction to Web1.0 and Web 2.0	10
	2.2 Rich Internet Applications (RIAs)	12
	2.3 Rich Internet Applications Web Client-Server architecture	13
	2.4 Summary	14
3	DESIGN AND DEVELOPMENT METHODOLOGIES	14
	3.1 AJAX based Applications for Web 2.0	17
	3.2 AJAX Open Standards	18
	3.3 Open Source as choice of AJAX development tools	19
	3.4 Faster Web Development using AJAX tools	21
	3.5 Analysis and Discussion on AJAX Architecture	22
	3.5.1 Client-side AJAX	22
	3.5.2 Server-side AJAX	23
	3.6 AJAX Source Library as Data Source	24

4	ANALYSIS AND DISCUSSION ON AJAX DEVELOPMENT	26
	4.1 Advantages of Using Client-Side JavaScript Library	29
	4.1.1 Disadvantages of Using Client-Side JavaScript Library	30
	4.2 Advantages of Using JavaServer Faces Library	32
	4.2.1 Disadvantages of Using JavaServer Faces Library	32
	4.3 Advantages of Using jMaki Library	35
	4.3.1 Disadvantages of Using jMaki Library	36
	4.4 Advantages and Disadvantages of using Direct Web Remoting	37
5	DEVELOPING A WEB CALENDAR SYSTEM USING APATANA WEB2 Integrated Development Environment (IDE)	39
	5.1 Data Source for testing and development using AJAX library	40
	5.1.1 Dojotoolkit Library	40
	5.1.2 Mochikit Library	42
	5.1.3 Prototype Library	43
	5.1.4 Open Rico Library	44
	5.1.5 Script.aculo.us Library	45
	5.1.6 Aflax Library	46
6	DEVELOPING WEB CALENDARING SYSTEM USING AJAX AND PHP (HYPERTEXT-PROCESSING LANGUAGE)	47
	6.1 Web 1.0 Calendar – Static and Unresponsive	47
	6.2 Web 2.0 AJAX Web Calendaring Application	48
	6.3 Web 2.0 AJAX Interactive validation	49
	6.4 Database creation for Web Calendar users	50

7	ANALYSIS AND DISCUSSION WEB CALENDARING SYSTEM	51
8	CONCLUSION	52
	REFERENCES	55
	APPENDIX I	58
	APPENDIX II	59
	APPENDIX III	62



## **ACKNOWLEDGEMENT**

I would like to thank Prof Dr Kasiran Buang for his wisdom and advise in the supervision of this work project paper. I would like to also thank him for the support throughout this work Master's course.

## **ABSTRACT**

This project consists of two parts. First part is to compare the Web 1.0 and Web 2.0 Internet company models. This study will help create an understanding of why a new development framework is recommended to create Web 2.0 application.

The second part is to use the new development framework called AJAX (Asynchronous JavaScript and eXtended Markup Language (XML)) to create a more interactive website based on Web2.0 model.

## DECLARATION


Name: Alex Lee Hong Sin

Matric. Number : CGS00031803

I hereby declare that this project paper is the result of this work own work, except for all quotations and summaries which have been acknowledged.

Any use of external authors, websites and books whether physical or electronic are declared in the references.

Signature: \_\_\_\_\_



Date: \_\_\_\_\_

5/6/07

# **1 INTRODUCTION**

This research project was undertaken to investigate on the various technology that are available in the market which are used in the development of websites and applications based on Web2.0 model for Internet based companies.

In addition, AJAX development technology has been chosen as the tools to build an application that is approaching the quality of desktop application, which is interactive and fast compared to the old technology used in the Web1.0 model.

## **1.1 Background of the Study**

This research project consists of two parts. The first part is about introducing web 2.0 as an internet business model of successful Internet-based companies.

The second part will cover the web development framework used by some of the web 2.0 Internet-based companies to drive its operations and create an environment for the Internet citizens to create, contribute, share and store ideas and knowledge in World Wide Web space.

Comparing internet business model from Web 1.0 model to Web 2.0 model. This research on Web 2.0 will cover basic awareness about Web 1.0 and Web 2.0. Find the differences among the two models, in areas of applications and web development technologies.

Understanding how AJAX works and how it can be used to overcome disadvantages and improve on the use of Internet will also be discussed.

The entire project development uses Open Source Tools such as Aptana (Integrated Development Environment) for client interface and PHP Programming Language for server side application.

## **1.2 Problem Statement**

Web 1.0 is not suitable for developing interactive Web application. Web 2.0 has been introduced to solve this shortcoming. This research work will show that web application based on Web 2.0 is indeed more interactive.

### **1.3 Research Objectives**

The objectives of this work consist of:

- (a) To demonstrate Web 2.0 application is more interactive compared to Web 1.0 application.
- (b) To develop a prototype web application using Web 2.0 tools.

## **2 LITERATURE REVIEW**

The study of Web 1.0 and Web 2.0 companies which are successful are the first important step in researching what makes Web 1.0 and Web 2.0 companies unique and different. The Web 1.0 companies such as DoubleClick, Ofoto, Britannica Online, mp3.com, Personal websites, Directories (Taxonothis work), Screen Scraping are popular to web surfers to download files and search for information.

The Web 2.0 companies such as Google AdSense, Flickr, Wikipedia, Napster, Blogging, Tagging (Folksonothis work), Web Services are the new version companies that are popular for social networks, content sharing, advertising and providing web applications.

A web 1.0 company uses the following programming languages in the development of their websites. The Client-Server Interface programming languages are Common Gateway Interface (CGI), Perl and C language. Client development mostly using Hypertext Markup Language (HTML) and JavaScript. Server development mostly using (Java Server Pages) JSP or PHP (PreHypertext Processor). The frameworks used in Web 1.0 Company are mostly Java based web frameworks such as Struts, Java Server Faces (JSF) are scripting based frameworks.

### **2.1 Introduction to Web 1.0 and Web 2.0**

The Web 1.0 companies are not producing Rich Internet Applications (RIAs) at that time due to the web applications are using Plugins, Applets, Frames and Java iframes, dumb browser which need plugins to view multimedia content, server centric the need to always load the page causing network congestion and lastly page to page navigation.

Web 2.0, a phrase coined by O'Reilly Media in 2004, refers to a supposed second-generation of Internet-based services such as social networking sites, wikis, communication tools, and folksonomies that let people collaborate and share information online in previously unavailable ways. O'Reilly Media, in collaboration with MediaLive International, used the phrase as a title for a series of conferences and since then it has become a popular (though ill-defined and often criticized) buzzword amongst certain technical and marketing communities. (Source: "What is Web 2.0", Tim O'Reilly (2005-09-30), O'Reilly Network website)

In the article (Source: "Levels of the Game: The Hierarchy of Web 2.0 Applications", Tim O'Reilly, O'Reilly Radar website). He mentioned that Level-3 applications, the most "Web 2.0", which could only exist on the Internet, deriving their power from the human connections and network effects Web 2.0 makes possible, and growing in effectiveness the more people use them. O'Reilly gives as examples: eBay, craigslist, Wikipedia, del.icio.us, Skype, dodgeball, and AdSense.

Level-2 applications operate offline but gaining advantages from going online. Flickr, which benefits from its shared photo-database and from its community-generated tag database. Level-1 applications, also available offline but which gain features online. Writely is gaining group-editing capability online and iTunes (because of its music-store portion). Level-0 applications would work as well offline. MapQuest, Yahoo! Local, and Google Maps. Mapping applications using contributions from users to advantage can rank as level 2. Non-web applications like email, instant-messaging clients and the telephone.

Technologies used in the running of Web 2.0 model companies are Blogging, Syndication, Really Simple Syndication (RSS) / Atom, Wikis, Web Services (REST) and AJAX, Rich Internet Clients.

Services provided by Web 2.0 companies Flickr, BitTorrent, iTunes (Entertainment and pictures), Maps (Yahoo, Google), Wikipedia (Knowledge and Information), Gmail, AdSense (Email and online advertising), Yahoo & Google Services (Web Portal) (Source: Article from Sun Microsystems website on Web2.0 and AJAX, 2005)

Web 2.0 Rich Internet Applications are Blogs / Really Simple Syndication (RSS) / Atom, RSS – Really Simple Syndication, Atom is latest, IETF, Standard, Provide Syndicated Information through (Hypertext Transfer Protocol) HTTP, Blogs build on RSS/Atom Aggregation, Content Reuse, Caching, Strong Social Phenomenon (e.g. politics), Rome – a popular RSS/Atom library, Roller – Apache project donated by Sun, Runs on Apache Solaris 9 and lastly Blogs.sun.com

Proponents of the Web 2.0 concept say that it differs from early web development (retrospectively labeled *Web 1.0*) in that it moves away from static websites, the use of search engines, and surfing from one website to the next, towards a more dynamic and interactive World Wide Web. Others argue that later developments have not actually superseded the original and fundamental concepts of the WWW. Skeptics may see the term "Web 2.0" as little more than a buzzword; or they may suggest that it means whatever its proponents want it to mean in order to convince their customers, investors and the media that they have begun building something fundamentally new, rather than continuing to develop and use well-established technologies (Source: Web 3.0. *A List Apart*. Jeffrey Zeldman 2006, viewed 27-05-2006)

Proponents of the Web 2.0 approach believe that Web usage has started increasingly moving towards interaction and towards rudimentary social networks, which can serve content that exploits network effects with or without creating a visual, interactive web page. In one view, Web 2.0 sites act more as points of presence, or user-dependent web portals, than as traditional websites. They have become so internally complex that new Internet users cannot create analogous websites, but remain mere users of web services provided by specialist professional experts.

Access to consumer-generated content facilitated by Web 2.0 brings the web closer to Tim Berners-Lee's original concept of the web as a democratic, personal, and DIY (Do-IT-Yourself) medium of communication.

Some websites that potentially sit under the Web 2.0 umbrella have built new online social networks amongst the general public. Some of the websites run social software

where people work together. Other websites reproduce several individuals' RSS feeds on one page. Other ones provide linking between individual websites.

The syndication and messaging capabilities of Web 2.0 have fostered, to a greater or lesser degree, a tightly-woven social fabric among individuals. Arguably, the nature of online communities has changed in recent months and years. The meaning of these inferred changes, however, has pundits divided. (Source: O'Reilly's coverage of Web 2.0 as a service mark, O'Reilly, 2006)

## **2.2 Rich Internet Applications (RIAs)**

The richer user-experience afforded by Ajax has prompted the development of web-sites that mimic personal computer applications, such as word processing, the spreadsheet, and slide-show presentation. WYSIWYG wiki sites replicate many features of PC authoring applications. Still other sites perform collaboration and project management functions. Java enables sites that provide computation-intensive video capability. One of the best known sites of this broad class, Writely, was acquired by Google in early 2006.

Several browser-based "operating systems" or "online desktops" have also appeared. They essentially function as application platforms, not as operating systems *per se*. These services mimic the user experience of desktop operating-systems, offering features and applications similar to a PC environment. They have as their distinguishing characteristic the ability to run within any modern browser.

Recently, rich-Internet application techniques such as Ajax, Adobe Flash and Flex have evolved that can improve the user-experience in browser-based web applications. Flash/Flex involves a web-page requesting an update for some part of its content, and altering that part in the browser, without refreshing the whole page at the same time.

## **2.3 Rich Internet Applications Web Client-Server architecture**

The functionality of Web 2.0 rich Internet applications builds on the existing web server architecture, but puts much greater emphasis on back-end software. Syndication differs



only nominally from the methods of publishing using dynamic content management, but web services typically require much more robust database and workflow support, and become very similar to the traditional intranet functionality of an application server. Vendor approaches to date fall under either a universal server approach, which bundles most of the necessary functionality in a single server platform, or a web-server plugin approach, which uses standard publishing tools enhanced with API interfaces and other tools. The extra functionality provided by Web 2.0 depends on the ability of users to work with the data stored on servers. This can come about through forms in an HTML page, through a scripting language such as JavaScript, or through Flash or Java. These methods all make use of the client computer to reduce the server workload.

The first and the most important step (in one view) of the evolution towards Web 2.0 involves the syndication of website content, using standardized protocols which permit end-users to make use of a site's data in another context, ranging from another website, to a browser plugin, or to a separate desktop application. Protocols which permit syndication include RSS (Really Simple Syndication — also known as web syndication), RDF (as in RSS 1.1), and Atom, all of them flavors of XML. Specialized protocols such as FOAF and XFN (both for social networking) extend functionality of sites or permit end-users to interact without centralized websites. Due to the recent development of these trends, many of these protocols remain *de facto* (rather than formal) standards.

Web client-server communications using web communication protocols provide a key element of the Web 2.0 infrastructure. Major protocols include REST and SOAP. REST (Representational State Transfer) indicates a way to access and manipulate data on a server using the HTTP verbs GET, POST, PUT, and DELETE. SOAP involves POSTing XML messages and requests to a server that may contain quite complex, but pre-defined, instructions for the server to follow

In both cases, an API defines access to the service. Often this API is specific to the server, but standard web service APIs (for example, for posting to a blog) are also widely used. Most, but not all, communications with web services involve some form of XML

(eXtensible Markup Language). (Source: Website of Sun Microsystem on Web 2.0 and AJAX, 2005)

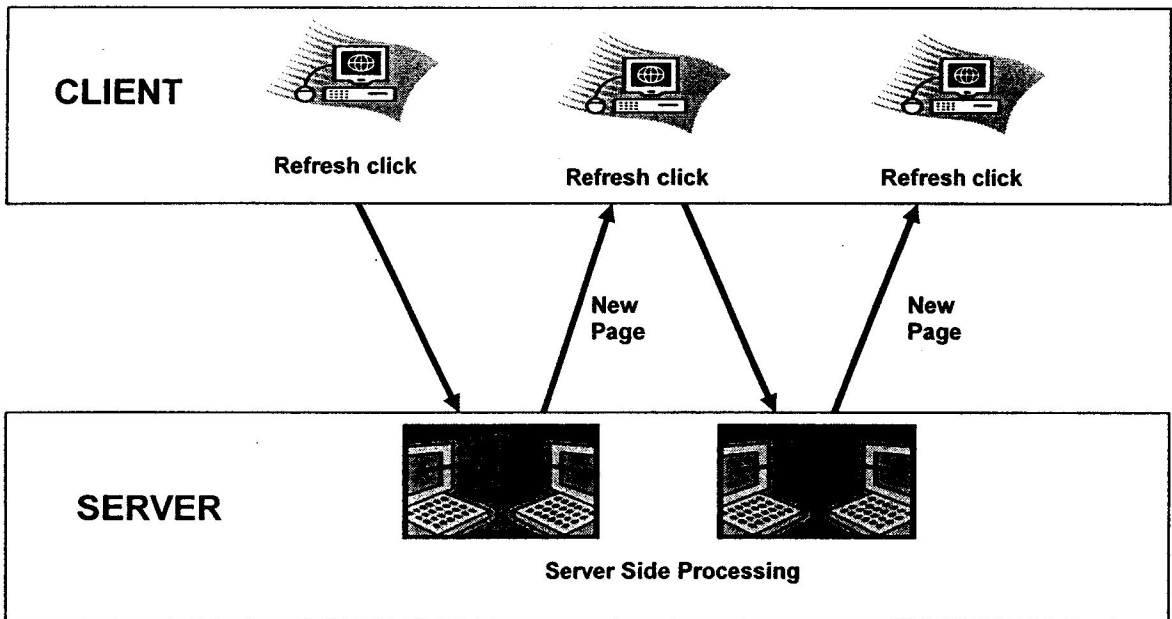
## **2.4 SUMMARY**

Given the lack of set standards as to what "Web 2.0" actually means, implies, or requires, the term can mean radically different things to different people. Many people will also talk about web sites "degrading gracefully" (designing a website so that its fundamental features remain usable by people who access it with software that does not support every technology employed by the site); however, the addition of Ajax scripting to websites can render the website completely unusable to anyone browsing with JavaScript turned off, or using a slightly older browser.

## **3 DESIGN AND DEVELOPMENT METHODOLOGIES**

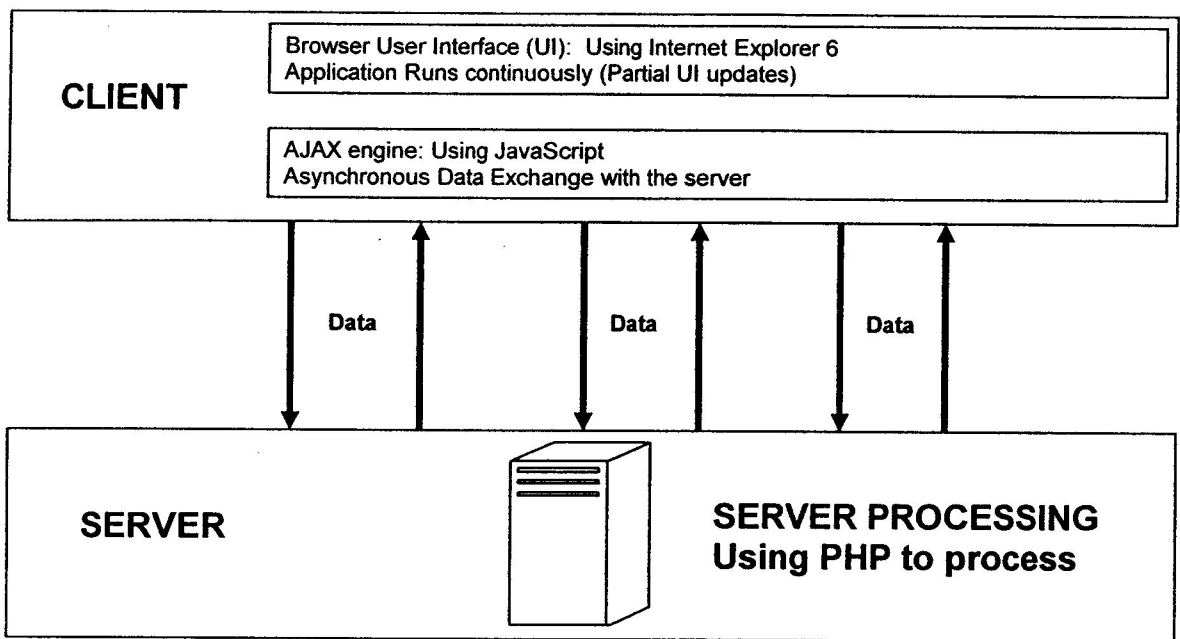
Ajax is a design approach and a set of techniques for delivering a highly interactive, desktop-like user experience for Web applications in popular HTML browsers. Ajax, which stands for Asynchronous JavaScript and XML, improves the user's web application experience while retaining the HTML benefits of server-based application deployment. Ajax represents the continued evolution of DHTML to deliver Web 2.0 experiences and Rich Internet Applications - RIAs.

Ajax builds on open standards that are available widely as native features (i.e., without plugins) in popular browsers. Ajax's incremental update techniques are accomplished through built-in features such as JavaScript and the XMLHttpRequest API. Ajax developments often leverage JavaScript libraries that provide a cross-browser abstraction layer. The key characteristics of Ajax: Dynamic and continuous user experiences, the primary difference between HTML applications and Ajax applications is that Ajax allows users to interact with the application while the browser is communicating with the server.



**Figure 1: Shows the typical "click, wait, and page refresh" user experience from typical HTML applications.**

Ajax minimizes the number of page refreshes because the client issues data requests to the server, not page requests. The Ajax application stays up on the screen continuously while the Ajax engine handles data exchange with the server. Instead of page refreshes, the application updates only those parts of the screen affected by newly arrived data.



**Figure 2: Using AJAX framework , shows there are no refresh of the page, but data are sent to the server when the client issues data request and not page requests**

Ajax enables rich user interfaces traditionally only found in installable desktop software. Typical rich user interface features found in Ajax applications include. Standard user interface (UI) controls (e.g., buttons, input fields and combo boxes), Advanced UI controls (e.g., tabbed palettes, calendar widgets, tree widgets and data grid widgets), Flexible, dynamic layout containers that adjust to the size of the embedded content and window size, Floating palettes and modal dialogs, Animations and animated effects and cut and paste, and drag and drop

Network programming features make it possible to exchange data asynchronously with the web server, enabling next-generation collaborative applications. Ajax runtime libraries often support the ability for servers to push data asynchronously to the client.

### **3.1 AJAX based Applications for Web 2.0**

Ajax offers a desktop-like user experience while retaining the benefits of server-based application deployment, centralized infrastructure administration, easier scalability and immediate availability of application updates to all users. As a result, Ajax is accelerating the movement away from installable, two-tier, client-server applications to multi-tier web applications. The runtime companion for Service-Oriented Architecture (SOA) is as the natural evolution of HTML, Ajax's platform-independent runtime technology is well-suited for next-generation service-oriented architecture (SOA) applications. Ajax offers standards compliance and platform independence on the client, while SOA offers similar benefits on the server. Ajax enables mixing and matching of multiple component technologies within the same application. This enables Ajax developers to build composite applications that leverage best-of-breed Ajax technologies from multiple suppliers, achieving many different types of composite applications.

**Mashups** - A mashup is a website or web application that uses content from more than one source to create a completely new service. One concrete example is a custom mapping application that talks to a company's own server to pull down XML address data and then leverages mapping services (e.g., Google or Yahoo) to show a map view in combination in an application-specific custom user interface. Mashups often enable rapid application development by integrating ready-made third-party components.

**Portals and dashboards** - Portals and dashboard applications consist of multiple panes that often can be configured by the user. Often, each pane is a separate application. Ajax technologies can be used to manage the entire composite application and its integrated applications. **Service compositions** - In an SOA environment, composite business applications assemble multiple application services to create new functionality that supports innovative business processes. In some cases, the services from which the new application is composed have no user interface, so the composite application is completely responsible for the user interface. In other cases the services do have user interfaces, so that the composite application knits together not only back-end behavior components, but user interface components as well.

Ajax enables collaborative applications, such as next-generation chat and instant messaging (IM) - Ajax enables web-based messaging applications that run in the browser. Web meetings and whiteboarding - Ajax's rich user interface, network programming, and vector graphics features provide the infrastructure for browser-based web meetings and shared whiteboards. Collaborative content creation - Web-based collaborative applications such as wikis can transition from users having to learn arcane wiki markup languages to a WYSIWYG user experiences, comparable to what users experience today with desktop content creation tools. Trip planning - A social group can use an Ajax-powered application to work together to plan and schedule their activities. Photo sharing - Ajax enables richer sharing of experiences, such as attaching annotations, tagging or marking up photos.

An important new application area is the Ajax-powered wiki, where wikis go beyond text-based collaborative documents into the following scenarios Rich, lightweight portals – Ajax-powered wikis enables fast deployment of Enterprise rich dashboards by allowing a wiki page to contain both text content and Ajax-powered user interface components, such as data grids, forms fillout, mapping, and charting. Personal mashups and dashboards – Many leading Internet companies provide the option today for customized home pages. When personalization is paired with Ajax-powered components and wiki back-end services, IT gains easy deployment and end-users gain the ability to manage their view on information.

### **3.2 AJAX Open standards**

Ajax leverages a combination of open technologies that are native to browsers. Most are official Web standards, while many of the rest have been implemented widely in browsers but have not been formally recognized by a standards body:

- **JavaScript** -- its official standards name is ECMAScript
- **HTML** -- often delivered via well-formed XHTML
- **CSS** -- for marking up and styling information

- **DOM and DOM Events** -- for client-side scripting APIs that interact with and control the web page
- **XMLHttpRequest** -- for client-server communications without requiring page refreshes (not yet an official standard, though under consideration by the W3C)
- **XML** -- a format for transferring data between the server and client (other formats can be used with Ajax, including preformatted HTML, plain text and JSON)
- **SVG** -- the standards-compliant vector graphics component for web pages (supported by recent releases of popular browsers; browsers that do not yet support SVG, Ajax libraries use other vector graphics support, such as VML in Internet Explorer)

Ajax builds on open standards that are widely available as native features (i.e., without plugins) in popular browsers. In most cases, Ajax restricts itself to the commonly implemented subset of particular standards (e.g., DOM2 and DOM3, which are not supported completely yet) or sometimes supports commonly implemented extensions (e.g., the innerHTML property within the DOM is implemented by most popular browsers but is not covered by W3C specifications).

### **3.3 Open source as choice of AJAX development tools**

While open source software is not mandatory for Ajax projects, a large part of Ajax's momentum is due to the open source community's commitment to Ajax. Today, many Ajax open source projects bring the power of community-based open development models and no-cost licensing models to developers. Here are some of today's Ajax open source projects: I have selected the Aptana (Integrated Desktop Environment(IDE)) to build this work Web 2.0 application. The Aptana IDE is an open source AJAX development tool and framework for Web2.0.

One of the main attractions of Ajax is that it does not lock developers to a particular hardware platform, operating system, application server, web browser or Integrated Development Environment (IDE). Developers are free to choose among many technology

providers, commercial and open source, to find the products or open source technologies that best match their unique requirements and preferences, while achieving the key benefit of write-once, run-everywhere, across multiple computing devices, operating systems, and web browsers.

Typically, Ajax toolkits deliver cross-platform and cross-browser functionality by providing a platform-neutral and browser-neutral abstraction layer to the developer. This layer is sometimes delivered as a set of client-side JavaScript libraries, and other times in the form of server-side software (e.g., Java).

Ajax can be added incrementally to existing HTML applications for a smooth and natural growth path to an Ajax-powered Web 2.0 and RIA user experience. Most of the technology behind Ajax is already familiar to the large pool of web developers who already know HTML and JavaScript. It is easy for them to learn quickly how to leverage Ajax to deliver next-generation solutions.

Ajax is fully compatible with the HTML application development infrastructure that exists today, including application servers (e.g., J2EE and .NET), server scripting languages (e.g., ASP, JSP and PHP), server application frameworks (e.g., JSF and Struts), web services, and service oriented architecture (SOA).

Organizations have the option of moving from HTML to Ajax in a phased manner. The following are the AJAX implementation strategies:

1. Add snippets of Ajax code within an HTML application
2. Use Ajax for the entire UI for one or more pages within a larger, primarily HTML web application
3. Use Ajax for the entire web application
4. Use Ajax as the basis for all web application development within your organization



### **3.4 Faster Web Development using AJAX tools**

With Ajax, the learning curve is shortened and investments minimized since application execution relies on Open standards support in web browsers. As a result, existing development and deployment frameworks and techniques still apply. The developer works in an environment he knows well and keeps full visibility and control, with the ability to code and debug all the way down to the Document Object Method (DOM) calls that affect what the browser displays. But Ajax also provides productivity advantages. Ajax toolkits typically offer declarative markup and APIs at higher abstraction levels and take care of lower-level details automatically.

The result is that Ajax offers the best of both worlds: automation advantages while still leaving the developer fully empowered.

With so many Ajax commercial products vendors and open source initiatives, developers are likely to find the off-the-shelf components, toolkits, frameworks, educational materials, and other resources they need to deliver and maintain next-generation Web 2.0 applications built with Ajax.

Many Ajax technologies provide declarative options (HTML/XML) for defining large parts of an Ajax application. These declarative options often automate large parts of the application development process and enable better leverage of Integrated Desktop Environment (IDE).

Some Ajax libraries deliver various JavaScript-oriented application development convenience features, such as JavaScript packaging and debugging aids. Some Ajax libraries go even further and deliver full application development platforms, sometimes in conjunction with associated software products such as IDEs. IDEs sometimes provide both server-side and client-side debugging.

3.5 ANALYSIS AND DISCUSSION ON AJAX Architectures

Ajax developers have the ability to choose among multiple technical approaches to find the ones whose programming model best fits their needs. The following are some of the ways to categorize Ajax toolkits and frameworks.

Most Ajax technologies transform a platform-independent definition of the application (usually consisting of a combination of markup and logic) into the appropriate HTML and JavaScript content that is then processed by the browser to deliver a rich user experience. Some Ajax designs perform most of their transformations on the client. Others perform transformations on the server.

3.5.1 Client-side Ajax transformations

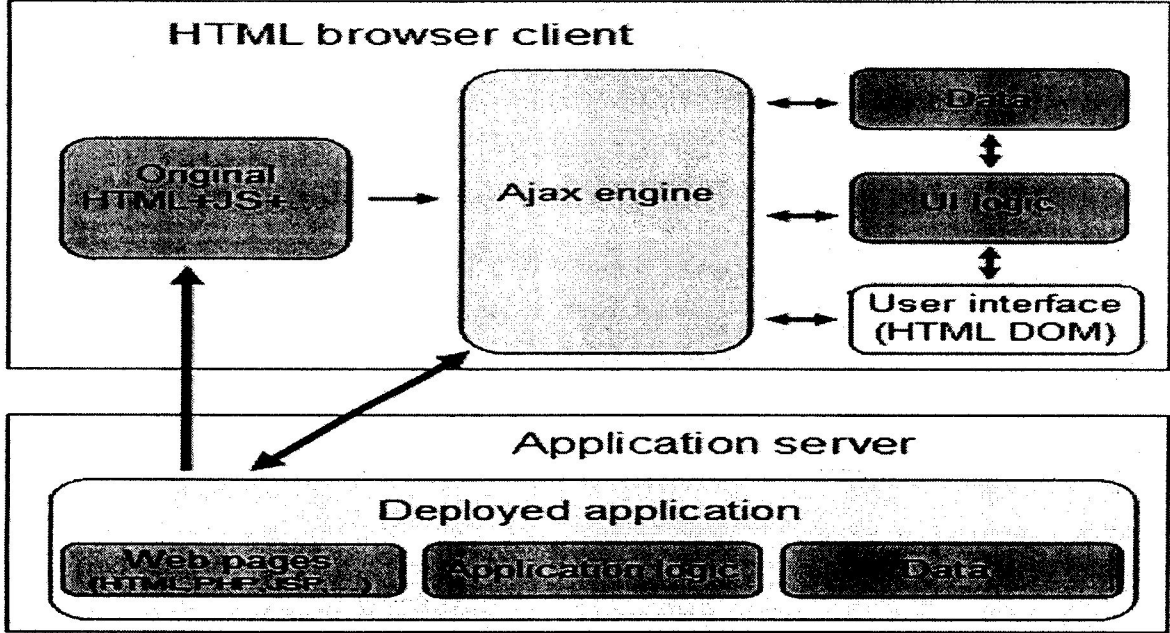


Figure 3: Client-side Ajax model – Successful Deployment of Ajax and OpenAjax  
(Source:<http://www.openajax.org/whitepapers/>)

The developer provides the green-color components, the Ajax toolkit provides the orange-colored components, and the Ajax engine generates the yellow-colored components, such as the HTML DOM, as a result of its transformations.

For client-side Ajax, the application server usually requires no additional Ajax-specific server software. All Ajax transformations come from the client-side Ajax runtime library.

One advantage of this option is the independence from the server side technology. The server code creates and serves the page and responds to the client's asynchronous requests. This way either side can be swapped with another implementation approach.

3.5.1 Server-side transformations

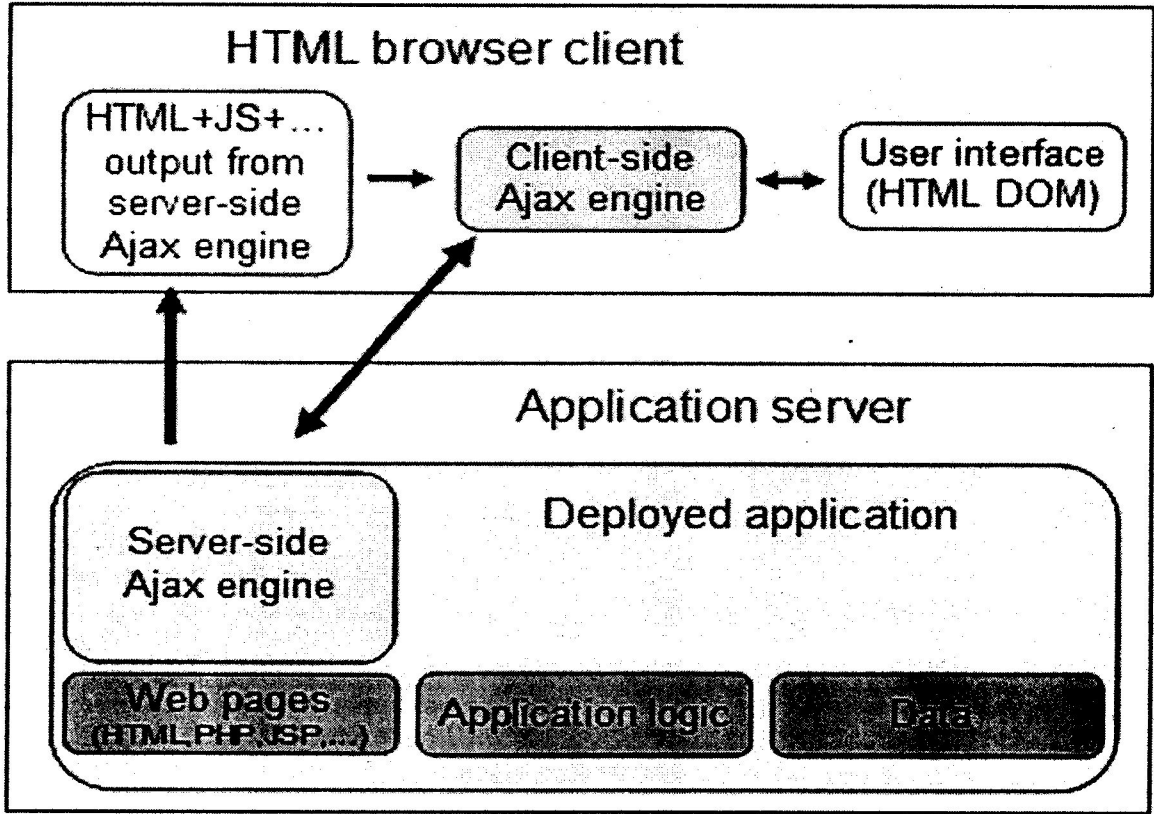


Figure 4: Server-side AJAX model – Successful Deployment of Ajax and OpenAjax  
(Source: <http://www.openajax.org/whitepapers/>)

For server-side Ajax, an Ajax server component performs most or all of the Ajax transformations into appropriate client-side HTML+JavaScript. Often, the server-side Ajax toolkit downloads its own client-side Ajax library which communicates directly with the toolkit's server-side Ajax component.

The main benefit of this approach is that it allows the use of server-side languages for debugging, editing, and refactoring tools with which developers are already familiar, but at the price of dependence on a particular server-side technology. As a general rule, server-side Ajax frameworks expect application code to be written in the server-side language (e.g. Java or RoR). These frameworks typically hide all the JavaScript that runs in the browser inside widgets, including their events. If pre-built capabilities don't suffice, however, new component development expose the programmer to JavaScript. The implementation strategies of server-based Ajax frameworks vary. At one side of the spectrum, the server handles all the application events. On the other side of the spectrum, many events are handled on the client. For some frameworks, the development and debugging phase handles all events on the server, but in production mode many events are handled on the client.

### **3.6 AJAX SOURCE LIBRARY AS DATA SOURCE**

The following are examples of AJAX source library that are used in the development of the Web Calendaring System. Dojo Toolkit is the Open Source JavaScript toolkit that helps you build serious applications in less time. It fills in the gaps where JavaScript and browsers don't go quite far enough, and gives you powerful, portable, lightweight, and tested tools for constructing dynamic interfaces. Dojo lets you prototype interactive widgets quickly, animate transitions, and build Ajax requests with the most powerful and easiest to use abstractions available. These capabilities are built on top of a lightweight packaging system, so you never have to figure out which order to request script files in again. Dojo's package system and optional build tools help you develop quickly and optimize transparently. AjaxWrite is a web-based word processor that can read and write Microsoft Word and other standard document formats. Google Web Toolkit (GWT) is a Java software development framework that makes writing AJAX applications like